UNIVERSIDAD DE MÁLAGA

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

SISTEMAS

ENTORNO PARA SCILAB; HACIA EL SOFTWARE LIBRE EN MÉTODOS NUMÉRICOS.

Realizado por

Álvaro Fernández Doblas

Dirigido por

D. Ángel Mora Bonilla

Departamento

Matemática Aplicada

UNIVERSIDAD DE MÁLAGA

MÁLAGA, (Octubre-2008)

5.3 Manual de Instalación y uso de Scilab

5.3.1 Instalación

5.3.1.1 Introducción

El programa Scilab UMA es usado para realizar todo tipo de cálculo numérico, así como para derivadas simbólicas y graficas, se dejará para su descarga en MetNum, el curso virtual de la asignatura de Métodos Numéricos que imparte el director del proyecto.

5.3.1.2 Requerimientos

Para instalar y ejecutar Scilab UMA en un equipo este debe cumplir los siguientes requisitos mínimos:

Sistema Operativo: Windows XP, 2000, Vista

Espacio libre en HDD: 200 MB

5.3.1.3 Instalación bajo Windows

Una vez obtenido el ejecutable, procedemos a la instalación. Para ello hacemos doble click en el archivo Scilab UMA.msi o el icono

Nos saldrá una serie diálogos que nos ayudarán a lo largo de este proceso.

Scilab Uma - InstallShield Wizard
Bienvenido a Scilab Uma - InstallShield Wizard
InstallShield(R) Wizard instalará Scilab Uma en su equipo. Para continuar, haga clic en Siguiente.
ADVERTENCIA: Este programa está protegido por las leyes de derechos de autor y otros tratados internacionales.
< Atrás
Siguiente >
Cancelar

Ilustración 48 (Instalación SCILAB UMA: Paso 1)

En el siguiente paso especificamos la ruta donde queremos que se instale la aplicación.

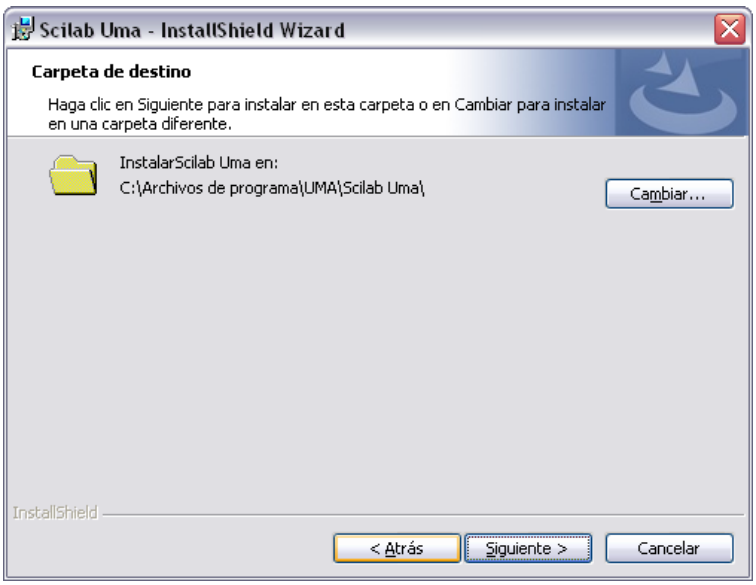

Scilab Uma - InstallShield Wizard
Carpeta de destino
Haga clic en Siguiente para instalar en esta carpeta o en Cambiar para instalar en una carpeta diferente.
InstalarScilab Uma en:
C:\Archivos de programa\UMA\Scilab Uma\
Cambiar...
InstallShield
< Atrás
Siguiente >
Cancelar

Ilustración 49 (Instalación SCILAB UMA: Selección de ruta para instalación)

Posteriormente pulsamos sobre el botón **instalar** para continuar.

Scilab Uma - InstallShield Wizard
Preparado para instalar el programa
El Asistente está preparado para comenzar la instalación.
Haga clic en Instalar para comenzar la instalación.
Si desea revisar la configuración de la instalación o realizar algún cambio, haga clic en Atrás. Haga clic en Cancelar para salir del Asistente.
InstallShield
< Atrás
Instalar
Cancelar

Ilustración 50 (Instalación SCILAB UMA: Instalar)

Una vez completada la instalación clicando sobre el botón **Finalizar**, saldremos del wizard.

5.3.2 El Programa Scilab UMA

El interfaz grafico Scilba Uma, tiene como objetivo aunar el motor matemático Scilab, la potencia de Maxima para la realización de derivadas simbólicas y proporcionar al usuario un entorno de trabajo de fácil manejo sin quitarle eficacia ni flexibilidad.

Ha sido desarrollado como proyecto de fin de carrera y su objetivo principal son los alumnos de la asignatura Métodos Numéricos que deseen utilizar una alternativa de código libre a Matlab.

Para iniciar la aplicación, como cualquier otra aplicación de Sistemas Windows, podemos hacerlo clicando dos veces sobre el icono correspondiente o por medio del menú de inicio. Al arrancar el programa se abre una ventana del tipo que se indica en la Ilustración 51.

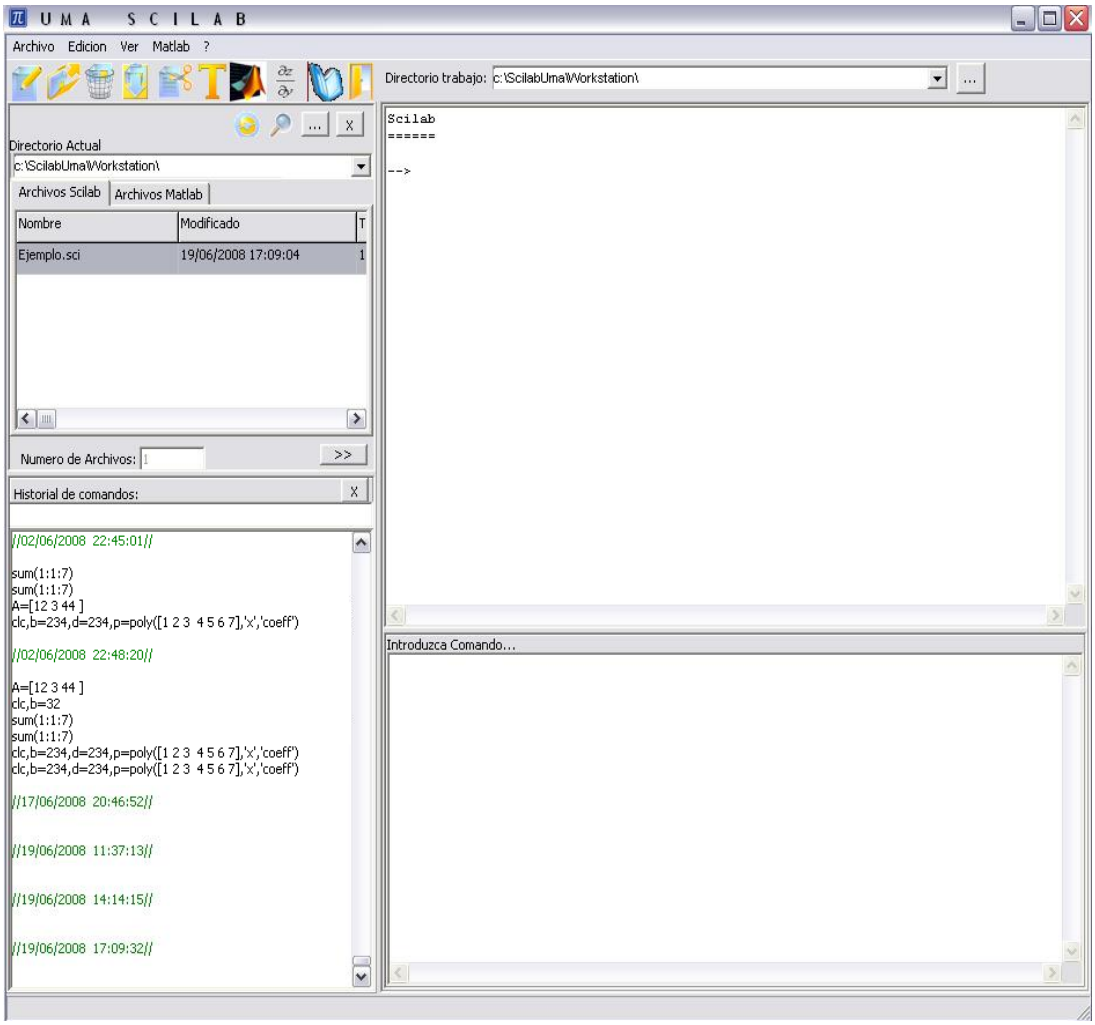


Ilustración 51 (Uso SCILAB UMA: Configuración inicial de Scilab UMA)

5.3.3 Espacio de trabajo de Scilab UMA

El entorno de trabajo Scilab UMA es muy grafico e intuitivo. Este entorno se divide en tres partes principales:

- *Ventanas de comandos*, que se utilizarán para interactuar con el motor matemático Scilab.
- *Historial de comandos*, en esta ventana podemos acceder a las sentencias que se hayan ejecutado previamente en la *ventana de comandos*.
- *Ventana de directorio activo*, en donde se muestran los ficheros con la extensión .sci o .sce y .m, que se encuentren en el directorio de trabajo actual.

5.3.3.1 Ventanas de Comandos.

En esta ventana (Ilustración 52) es donde se ejecutan de forma interactiva las instrucciones de Scilab (sub-ventana para introducir los comandos a ejecutar o *consola de comandos*, Ilustración 53) y donde se muestran los resultados correspondientes (sub-ventana salida de comandos o *consola de resultados*, (Ilustración 54), si es el caso.

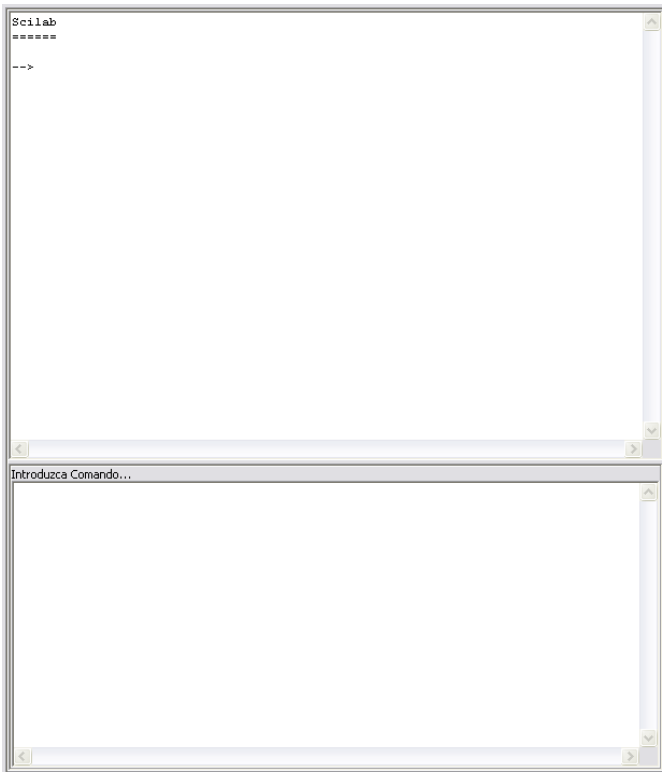


Ilustración 52 (Uso SCILAB UMA: Ventanas de comandos)

Ilustración 53 (Uso SCILAB UMA: Consola de Resultados)

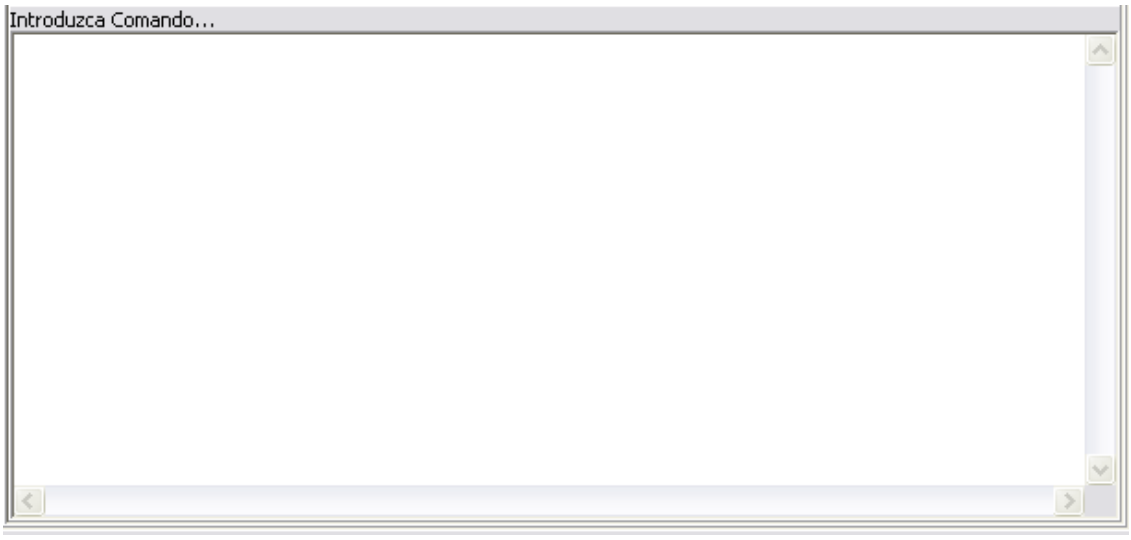


Ilustración 54 (Uso SCILAB UMA: Consola de comandos)

En la *consola de resultados* (Ilustración 53) podemos hacer click con el botón derecho del ratón y se nos mostrará un submenú (Ilustración 55) con una serie de opciones que se describen a continuación.

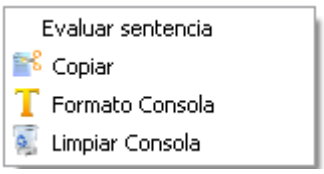


Ilustración 55 (Uso SCILAB UMA: Popup-menú ventada de comandos)

a) Evaluar sentencia. Con esta opción se nos permite re-evaluar cualquier sentencia enviada previamente al Scilab.
b) Copiar. Copiamos la salida obtenida, para pegarla posteriormente en la sub-ventana donde se introducen los comandos (Ilustración 54).
c) Formato consola. Podemos cambiar el tipo y el tamaño de la letra de *consola de resultados*.
d) Con la opción Limpiar Consola, eliminamos el contenido de la ventana.

5.3.3.2 Historial de Comandos.

La ventana *historial de comandos* (Ilustración 56) se encuentra situada en la parte inferior izquierda y en ella se muestran los comandos ejecutados anteriormente.

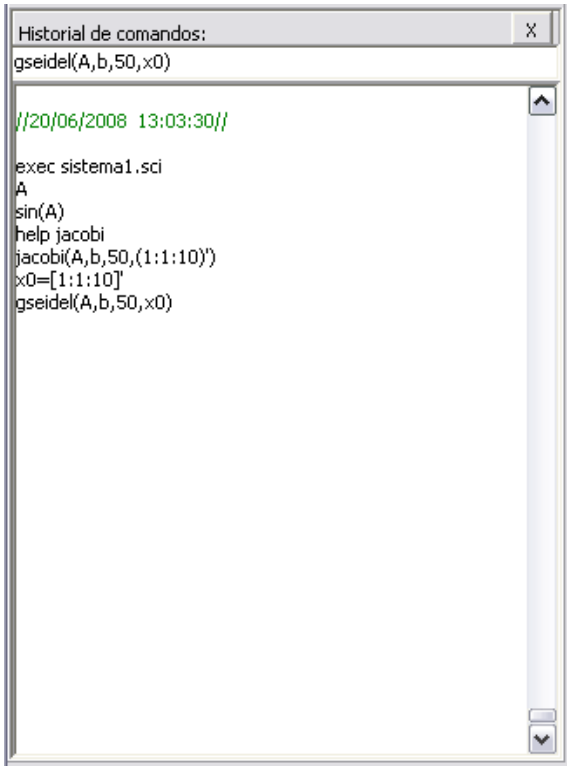


Ilustración 56 (Uso SCILAB UMA: Historial de comandos)

En ella, haciendo doble click sobre la sentencia, ejecutarla de nuevo. También tenemos la opción de seleccionar varios comandos (usando la tecla *control* y haciendo click sobre cada uno de ellos) y copiar o arrastrar a la ventana *consola de comandos*, para editarlos o evaluarlos de nuevo.

Cada vez que cerramos la aplicación se creara un separador en el cual se indica la fecha y hora de la última sesión activa.

En el *historial de comandos*, también encontramos una herramienta para hacer una búsqueda rápida de los comandos almacenados (Ilustración 57). Una vez encontrado, pulsando la tecla *enter*, se copiará a la *consola de comandos.*

Ilustración 57 (Uso SCILAB UMA: Búsqueda rápida de comandos)

En cualquier momento podemos borrar un comando o bien la lista entera de comandos. Para eliminar una sentencia en particular, nos situamos sobre ella, cliqueando con el botón derecho del rato elegimos la opción **Borrar** (Ilustración 58).

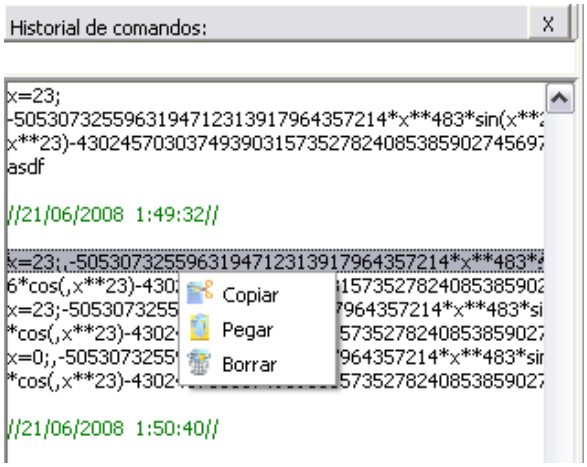


Ilustración 58 (Uso SCILAB UMA: Borrado de una sentencia del historial de comandos)

Si por el contrario queremos borrar todos los comandos, podemos ir al *Toolbar Menú*. y pulsar sobre o usar el menu principal **Edicion/Limpiar Historial de Comandos** (Ilustración 59).

Ilustración 59 (Uso SCILAB UMA: Menú Edición)

También podemos ocultar el *historial de comandos*, pulsando sobre el botón o bien utilizar el menú **Ver/Historial de comandos** (Ilustración 60).

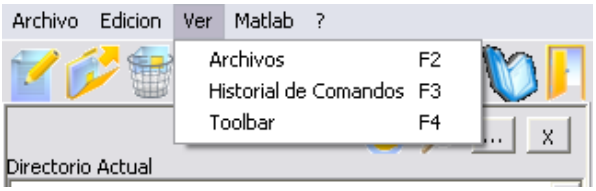


Ilustración 60 (Uso SCILAB UMA: Ocultación historila de comandos)

5.3.3.3 Ventana de directorio activo.

El concepto de directorio activo o actual es muy importante en SCILAB. Los programas en SCILAB tienen la extensión .sci o .sce, estos pueden contener o bien funciones o bien secuencias de comandos. Para ejecutarlos tenemos dos opciones que dependen del contenido de dichos ficheros:

- Usamos la función getf 'nombrefuncion.sci'. Si el fichero contiene una función.

  Ejemplo:

  ```
  function y=mifuncion(x)
      y=sin(x)+x^2-cos(x);
  endfunction
  ```

  ➔ getf 'mifuncion.sci'
- Usamos la función exec 'nombrefichero.sci'. En el caso de que el fichero nombrefuncion.sci, contenga un conjunto de sentencias.

  Ejemplo:

  (Contenido del fichero miarchivo.sci)

  ```
  A=[1 3 9 15 2.5 5];
  B=[0.2 1 5 4 8.5 1];
  A+B
  B-A
  ```

  ➔ Exec 'miarchivo.sci'

Tenemos que tener en cuenta que tanto para cargar como para ejecutar un fichero, tenemos que estar en el directorio donde se encuentra dicho fichero., ya que si no es así, se nos devolverá un error indicando que no se puede encontrar el archivo especificado.

En la ventana *directorio actual* (Ilustración 61), se nos muestra el directorio actual de trabajo, los ficheros con extensiones .sci, .sce y .m que se encuentran en dicho directorio, se nos permite hacer búsqueda de ficheros y cargar todas las funciones del directorio de trabajo.

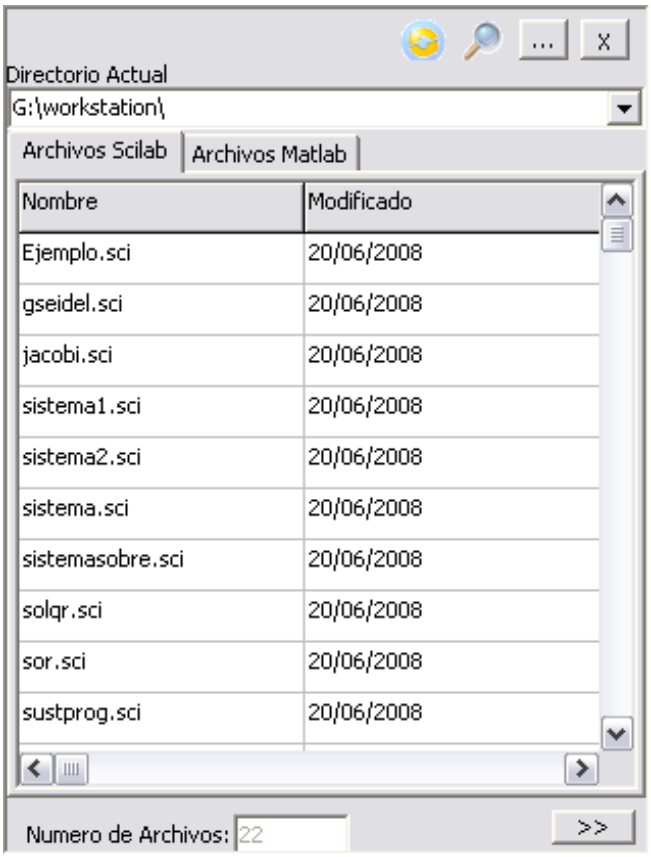


Ilustración 61 (Uso SCILAB UMA: Ventana de directorio actual)

En el desplegable *directorio actual* (Ilustración 62), podemos ver cuál es el directorio de trabajo. Tenemos que tener en cuenta que tanto para cargar como para ejecutar un fichero, tenemos que estar en el directorio donde se encuentra dicho fichero., ya que si no es así, se nos devolverá un error indicando que no se puede encontrar el archivo especificado.

Ilustración 62 (Uso SCILAB UMA: Directorio de trabajo)

Para cambiar de directorio de trabajo tenemos dos opciones:

1. Añadir nuevos Path
2. Especificar el directorio manualmente o usando el icono

Para añadir nuevos paths o rutas, tenemos que ir a **Archivos/Set Path…** o bien pulsar ***control+p***.

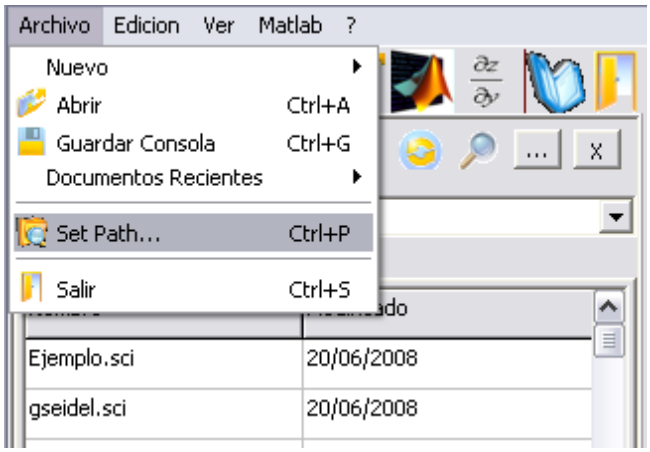


Ilustración 63 (Uso SCILAB UMA: Adición de nuevos path 1)

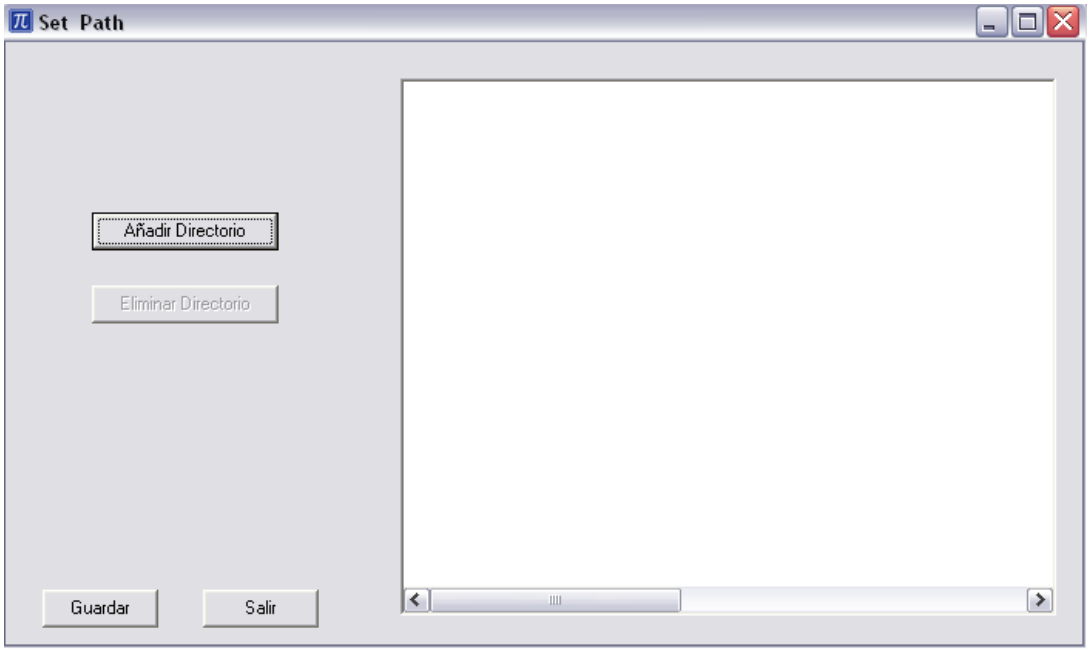


Ilustración 64 (Uso SCILAB UMA: Adición de nuevos path 2)

Una vez abierto el formulario, tendremos que pulsar sobre *Añadir Directorio*. Buscaremos el directorio deseado y le damos a Aceptar (Ilustración 65).

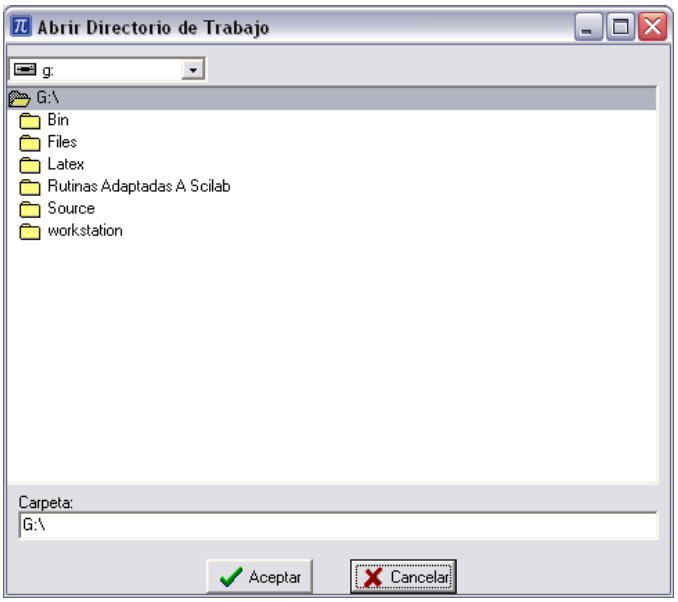


Ilustración 65 (Uso SCILAB UMA: Adición de nuevos path 3)

Es importante mencionar que para que se guarde el nuevo directorio, tenemos que pulsar sobre el botón *guardar*, antes de salir. Si salimos sin guardar la ruta no se almacenará.

Podemos borrar cualquier ruta almacenada previamente, entrando en el formulario *Set Path…* (Ilustración 63) una vez allí seleccionar el directorio a borrar y pulsar sobre el botón *Eliminar directorio*.

Una vez añadida las rutas, aparecerán el desplegable de menú actual, pudiendo cambiar de directorio en cualquier momento. Al cambiar de directorio se cargaran automáticamente todas las rutinas y ficheros que se encuentren en él.

Nota: Para los archivos que no sean funciones, tenemos que o bien escribir sobre la *consola de comandos* exec nombrefichero.sci o bien hacer click con el botón derecho del ratón y pulsar sobre la opción *Ejecutar* (Ilustración 66).

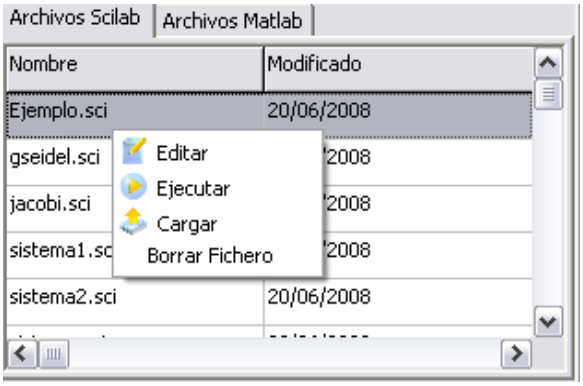


Ilustración 66 (Uso SCILAB UMA: Popup-menu ficheros del directorio de trabajo)

Para cambiar de directorio manualmente usaremos el botón . Buscamos el directorio deseado y le damos a aceptar (Ilustración 67). Cargando en memoria todos los ficheros que en él se encuentren.

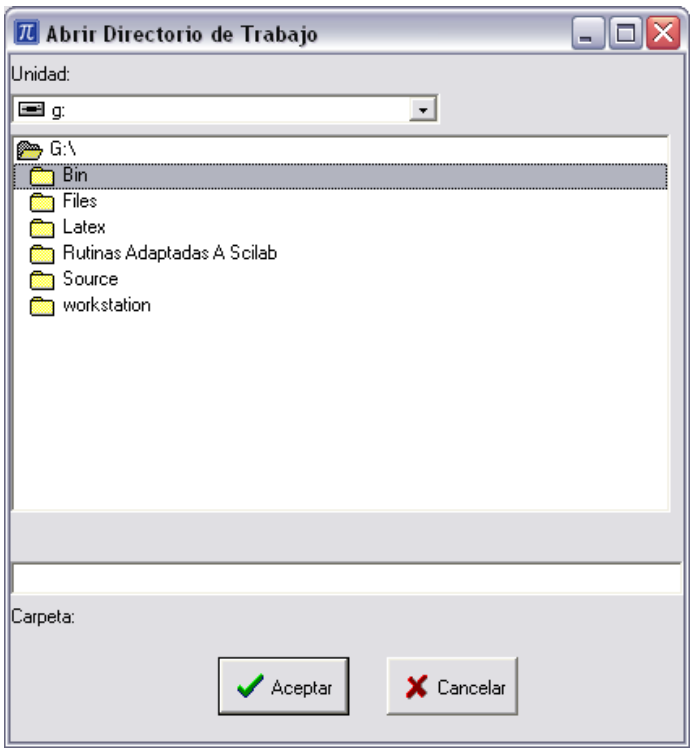


Ilustración 67 (Uso SCILAB UMA: Cambiar directorio de trabajo manualmente Paso 1)

En el caso de haber añadido nuevos ficheros en el directorio y que no aparezcan pulsaremos sobre el icono , recargando automáticamente el directorio y mostrándose todos los ficheros nuevos y antiguos. Para que los nuevos archivos se carguen debemos pulsar sobre el icono y se recargaran tanto los antiguos como los nuevos o los modificados.

Otra opción es seleccionar el fichero nuevo y usando el botón derecho del ratón hacer click sobre cargar o ejecutar dependiendo de si es un fichero de comandos o un fichero que contiene una función (Ilustración 68).

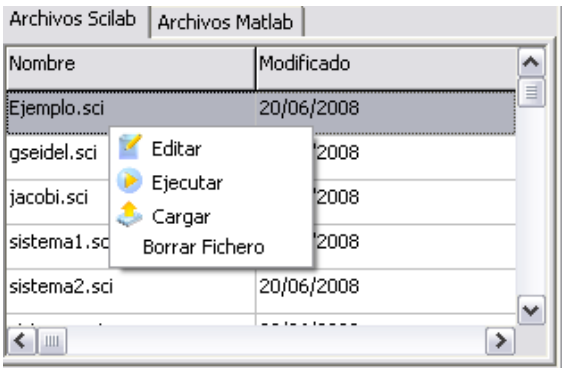


Ilustración 68 (Uso SCILAB UMA: Popup-menu de los ficheros para cargarlos y ejecutarlos)

Este menú (Ilustración 68) aparece siempre que nos encontremos en la pestaña de *Archivos Scilab*, si por el contrario estamos en la pestaña *Archivos Matlab*, el menú emergente cambiara de opciones (Ilustración 69). Estas pasarán a ser las siguientes:

- Editar: Para modificar el fichero seleccionado.
- Traducir a Scilab: Podemos traducir el fichero para usar las funciones de Scilab.
- Borrar Fichero: Mandaremos a la papelera el archivo seleccionado.

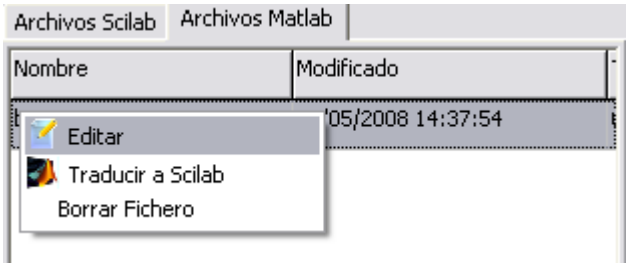


Ilustración 69 (Uso SCILAB UMA: Popup-menu opciones de *Archivos Matlab)*

Para realizar una búsqueda de ficheros, tenemos que pulsar sobre el icono se nos presentará un nuevo formulario (Ilustración 70) en el cual introducimos los datos del fichero o ficheros a buscar, así como del directorio donde se quiere realizar la búsqueda.

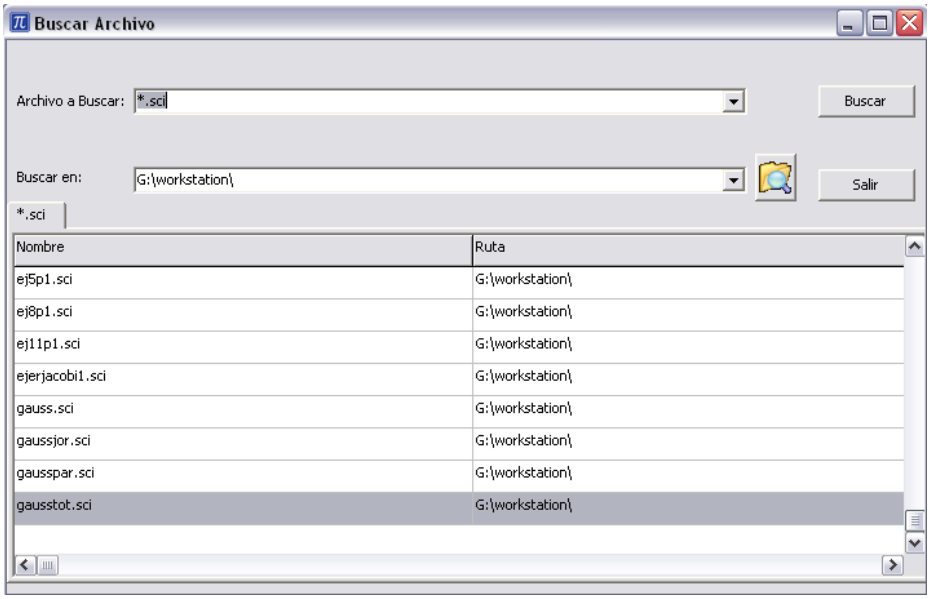


Ilustración 70 (Uso SCILAB UMA: Búsqueda de ficheros)

Al igual que con la ventana *historial de comandos*, podemos ocultar la ventana de directorio actual cliqueando sobre el icono , pulsar sobre la tecla **F2** o bien usando el menú principal de la aplicación **Ver/Historial de comandos** (Ilustración 60).

5.3.3.4 Toolbar Menú.

El *toolbar Menú* (Ilustración 71) es un menú que nos permite acceder de forma rápida a las principales opciones del programa.

Ilustración 71 (Uso SCILAB UMA: Toolbar Menú)

Las opciones presentes son las siguientes:

ICONO	FUNCION	DESCRIPCION
	Editar	Abriremos el editor de Scilab (scipad) para crear un nuevo documento de Scilab.
	Abrir Archivo	Podemos abrir cualquier documento Scilab creado previamente.
	Borrar Historial de Comandos	Borraremos todos los comandos que hayamos acumulado.
	Pegar	Pegamos lo que tengamos en el portapapeles en la ventana *consola de comandos*
	Copiar	Con esta opción podemos copiar los comandos del *historial de comandos*, previamente seleccionados.
	Cambiar Formato	Cambiar el tamaño y color de la fuente de la *consola de comandos*, *consola de resultados* y/o del *historial de comandos*
	Traducir de Matlab a Scilab	Se nos presenta un formulario para traducir archivos que tengamos en formato MATLAB a formato compatible con SCILAB.
	Derivada Simbólica	Podemos derivar cualquier función de forma simbólica.
	Ayuda	Abriremos la ayuda de Scilab.

	Salir	Terminaremos la ejecución de la aplicación.
	Directorio de Trabajo	Directorio en uso.

5.3.4 Uso de la Ayuda.

SCILAB posee un potente help (Ilustración 72), en el que podemos encontrar la información que deseemos acerca tanto de las funciones que vienen incorporadas en el programa como de las que se hayan añadido posteriormente.

Para acceder al *help* existen varios métodos:

- Mediante la tecla F1
- En la *toolbar Menú* pulsando sobre el icono
- Mediante **?/Ayuda**
- Tecleando help en la *consola de comandos.*(Ilustración 54)

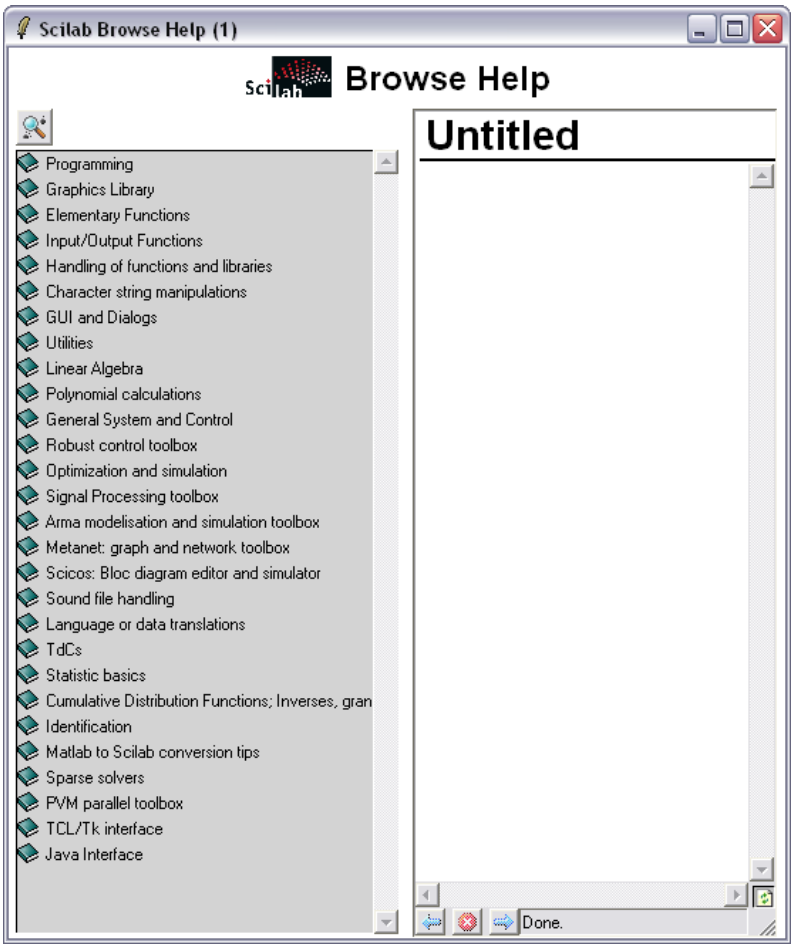


Ilustración 72 (Uso SCILAB UMA: Menú de ayuda)

Una vez abierto el *help*, podemos ir a la sección deseada y se nos desplegará todo lo relacionado a ella.

Así, si queremos ver todos lo relacionado la conversión de MATLAB a SCILAB, pulsamos sobre **Matlab to Scilab conversion tips**. Se nos mostrarán todas las funciones y una explicación de su uso (Ilustración 73).

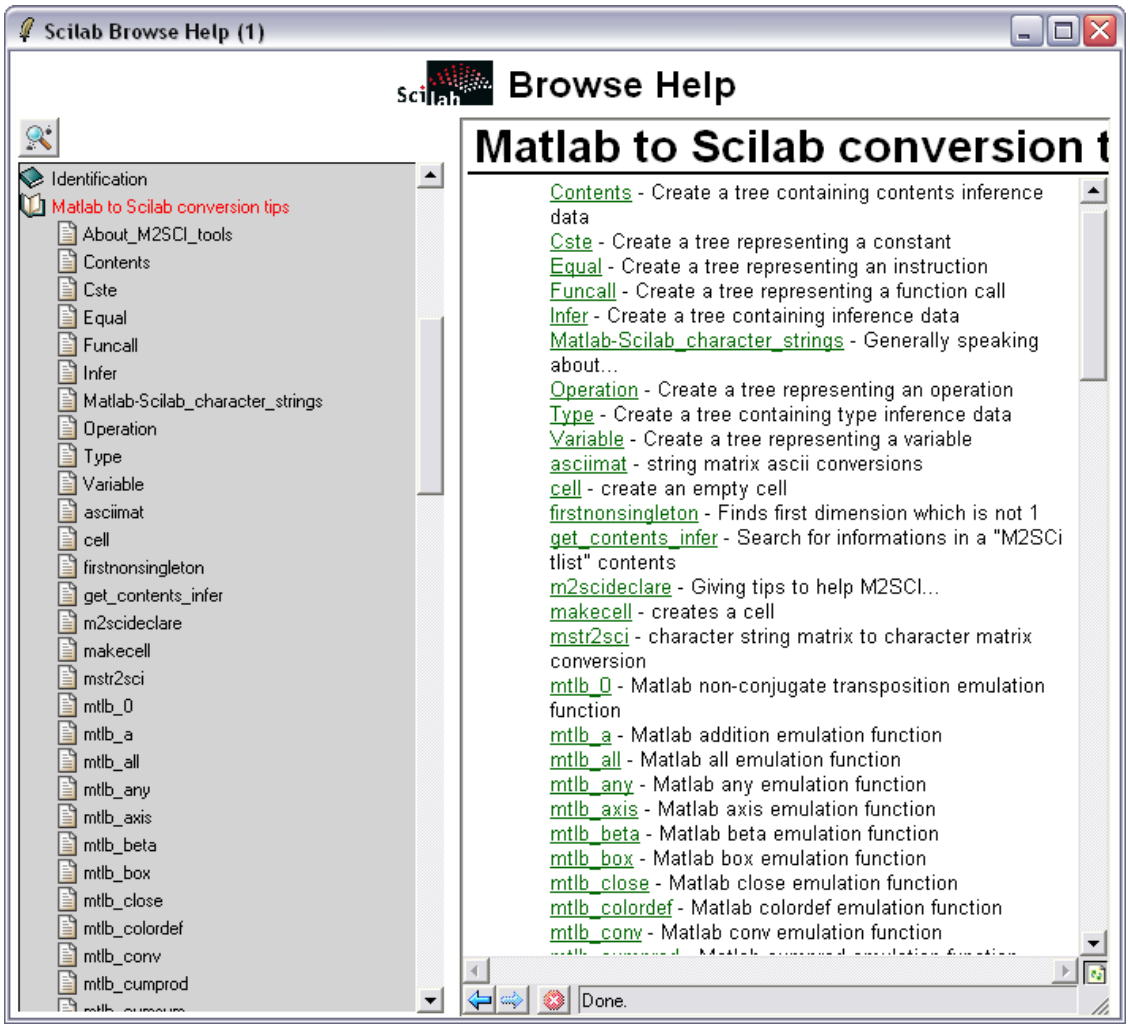


Ilustración 73 (Uso SCILAB UMA: Ayuda sobre la conversión desde Matlab a Scilab)

Si solo queremos buscar una función o variable o termino concreto, bastará con pulsar sobre el icono , y ahí introducimos el término a buscar (Ilustración 74).

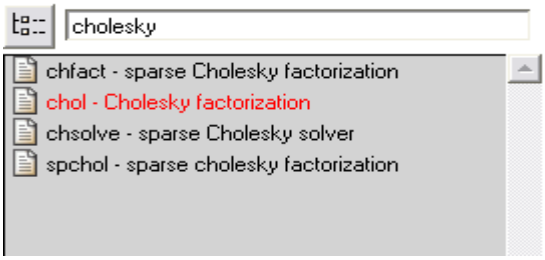

cholesky
chfact - sparse Cholesky factorization
chol - Cholesky factorization
chsolve - sparse Cholesky solver
spchol - sparse cholesky factorization

Ilustración 74 (Uso SCILAB UMA: Ejemplo de búsqueda en la ayuda)

Como ya se ha comentado anteriormente podemos realizar una búsqueda directa usando el comando **help**, desde la *consola de comandos.*

Si queremos buscar ayuda sobre el comando **poly**, teclearemos lo siguiente:

➔ help poly

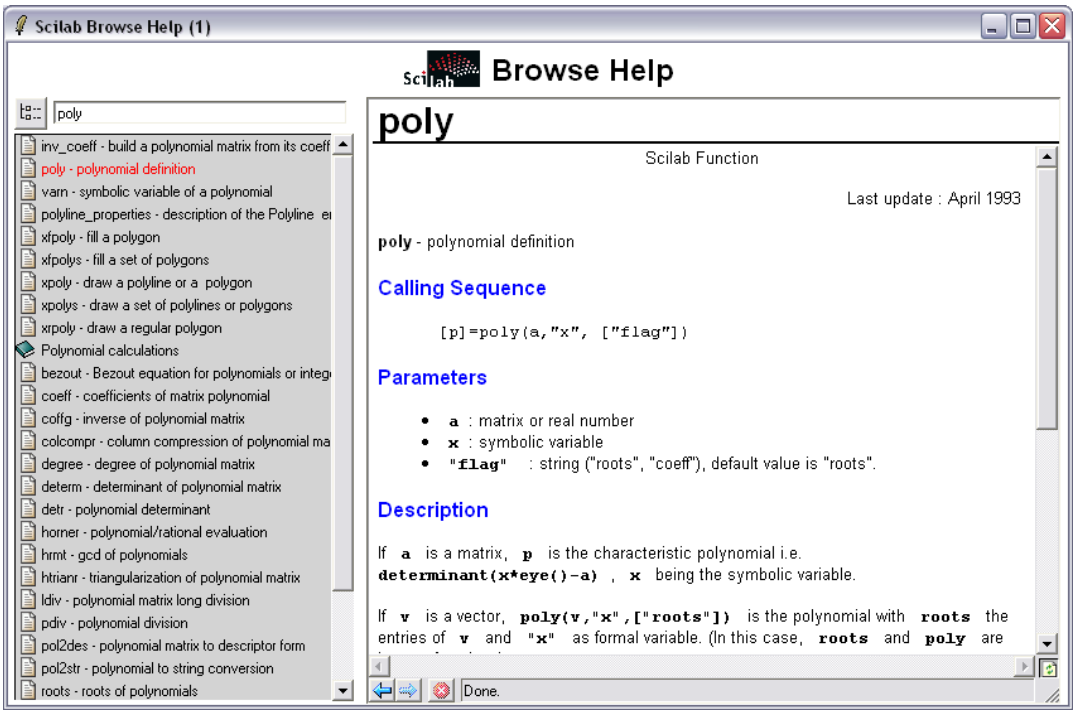

Scilab Browse Help (1)
Browse Help
poly
inv_coeff - build a polynomial matrix from its coeff
poly - polynomial definition
varn - symbolic variable of a polynomial
polyline_properties - description of the Polyline
xfpoly - fill a polygon
xfpolys - fill a set of polygons
xpoly - draw a polyline or a polygon
xpolys - draw a set of polylines or polygons
xrpoly - draw a regular polygon
Polynomial calculations
bezout - Bezout equation for polynomials or integ
coeff - coefficients of matrix polynomial
colfg - inverse of polynomial matrix
colcompr - column compression of polynomial ma
degree - degree of polynomial matrix
determ - determinant of polynomial matrix
detr - polynomial determinant
horner - polynomial/rational evaluation
hrmt - gcd of polynomials
htrianr - triangularization of polynomial matrix
ldiv - polynomial matrix long division
pdiv - polynomial division
pol2des - polynomial matrix to descriptor form
pol2str - polynomial to string conversion
roots - roots of polynomials
poly
Scilab Function
Last update : April 1993
poly - polynomial definition
Calling Sequence
[p]=poly(a,"x", ["flag"])
Parameters
a : matrix or real number
x : symbolic variable
"flag" : string ("roots", "coeff"), default value is "roots".
Description
If a is a matrix, p is the characteristic polynomial i.e.
determinant(x*eye()-a) , x being the symbolic variable.
If v is a vector, poly(v,"x",["roots"]) is the polynomial with roots the
entries of v and "x" as formal variable. (In this case, roots and poly are
Done.

Ilustración 75 (Uso SCILAB UMA: Ayuda sobre el comando poly)

5.3.5 Uso de la Derivación Simbólica.

La principal carencia de Scilab es la ausencia de una función para realizar derivación simbólica de funciones. Para mitigar esa carencia se ha añadido al entorno grafico un formulario que nos permite derivar simbólicamente una función con respecto a una o varias variables, tantas veces como se desee, y posteriormente evaluarla en un valor dado.

Para acceder a la derivación simbólica (Ilustración 76) podemos hacerlo desde el menú *toolbar* ($\frac{\partial z}{\partial y}$) o bien pulsando las teclas **control+d.**

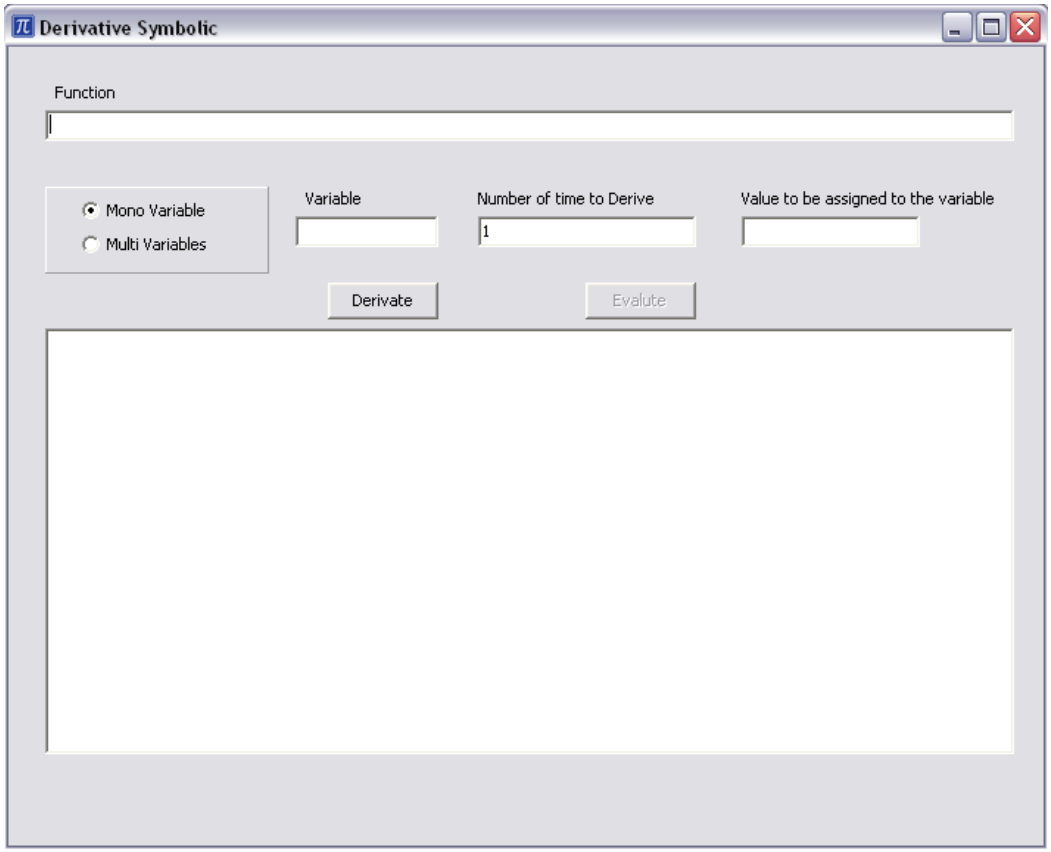


Ilustración 76 (Uso SCILAB UMA: Formulario para la derivación simbólica)

Donde:

- Función a Derivar: es la función a la queremos realizar la derivada.
- Seleccionamos si queremos hacerla derivación respecto a una o a varias variables, seleccionando *Mono Variable* o *Multi Variable*.
- Variable: el nombre de la variable sobre la que se derivará. En el caso de haber seleccionado la opción *Multi Variable*, introduciremos las variables deseadas separadas por ‘,’.
- Número de Veces a Derivar
- Valor de la variable: Si queremos evaluar la derivada en un punto, en este apartado especificamos dicho punto.

Si por ejemplo queremos derivar la función seno($x^2+e^{\cos(2x)}$, hacemos click sobre el icono $\frac{\partial z}{\partial y}$. Y por ultimo rellenamos el formulario con los datos:

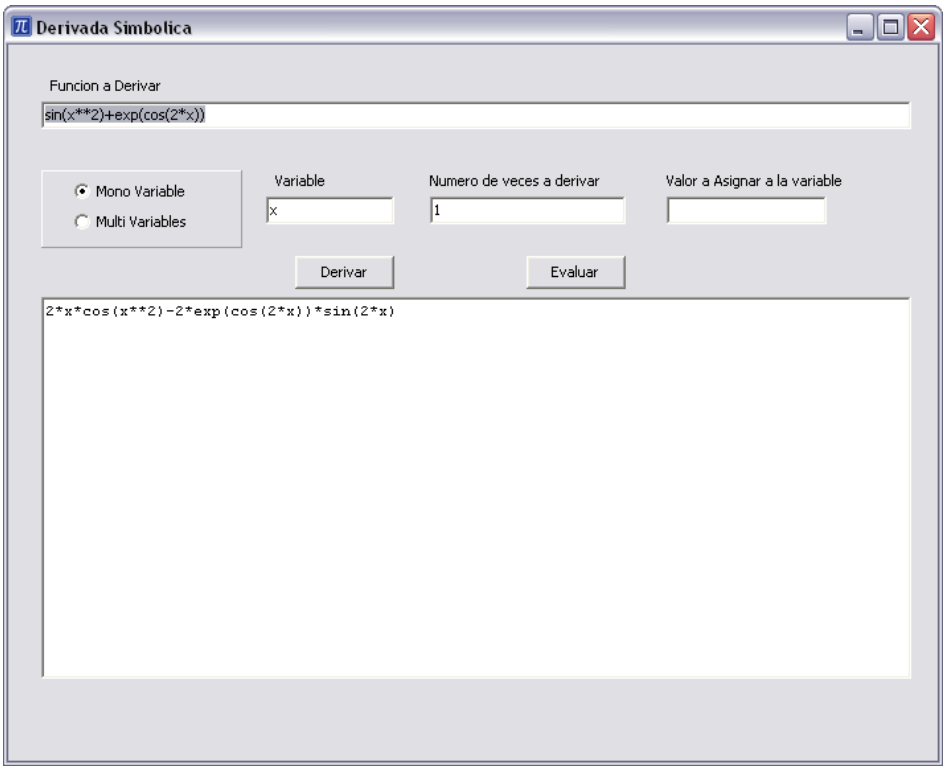


Ilustración 77 (Uso SCILAB UMA: Ejemplo de derivación simbólica)

Para evaluarla en el valor x=1.23, ponemos 1.23 en el campo **valor de la variable** y pulsamos sobre el botón **Evaluar**. Copiando el valor de la variable y la función en la *consola de comandos*. Pulsando la tecla **enter** se evaluará la sentencia (Ilustración 78).

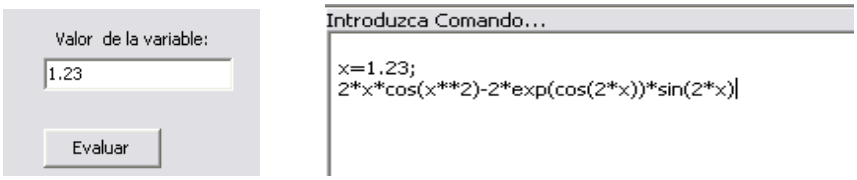
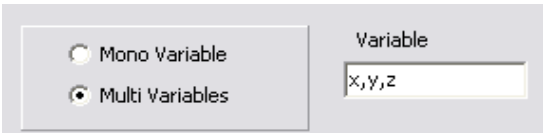


Ilustración 78 (Uso SCILAB UMA: Evaluar la derivada)

Si lo que queremos es realizar la derivación respecto de multiples variables, seleccionamos la opción *Multi Variable* y en apartado *Variables*, vamos poniendo las variables a derivar separadas por ‘,’.



5.3.6 Uso de La traducción de MATLAB a SCILAB

Muchas funciones que usa MATLAB (como eye(), ones(), size()…) se comportan de forma diferente en función de la dimensión de sus argumentos.

Si bien SCILAB, posee una serie de funciones que emulan las funciones de MATLAB, estas funciones tienen la siguiente forma mtlb_nombrefuncion. Para ver las funciones emuladas, yendo al *help*, hacemos la búsqueda con la cadena "mtlb_", dando como resultado todas las funciones de SCILAB que emulan las de MATLAB (Ilustración 79).

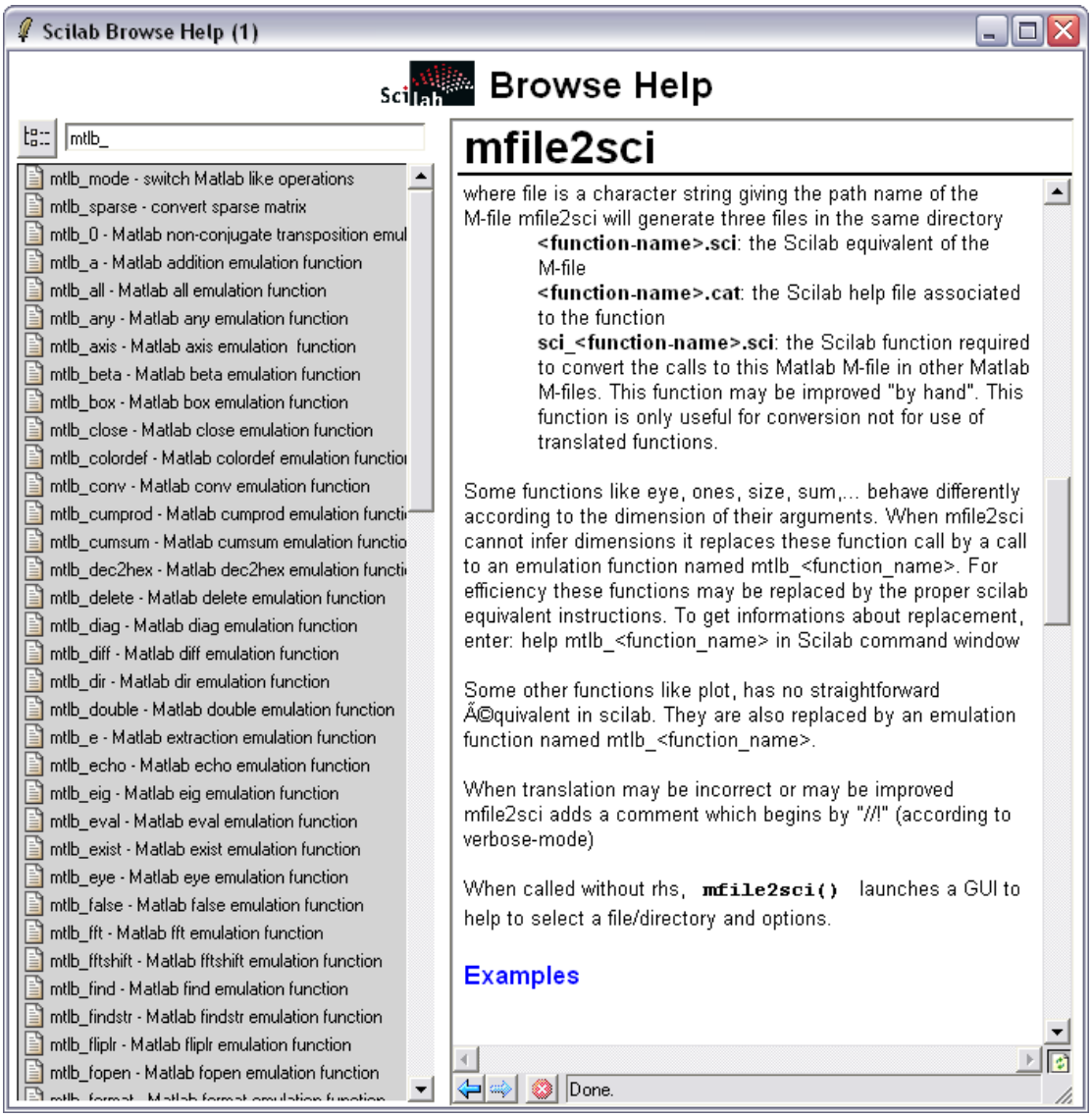

Scilab Browse Help (1)
Browse Help
mtlb_
mtlb_mode - switch Matlab like operations
mtlb_sparse - convert sparse matrix
mtlb_0 - Matlab non-conjugate transposition emul
mtlb_a - Matlab addition emulation function
mtlb_all - Matlab all emulation function
mtlb_any - Matlab any emulation function
mtlb_axis - Matlab axis emulation function
mtlb_beta - Matlab beta emulation function
mtlb_box - Matlab box emulation function
mtlb_close - Matlab close emulation function
mtlb_colordef - Matlab colordef emulation functio
mtlb_conv - Matlab conv emulation function
mtlb_cumprod - Matlab cumprod emulation functi
mtlb_cumsum - Matlab cumsum emulation functio
mtlb_dec2hex - Matlab dec2hex emulation functi
mtlb_delete - Matlab delete emulation function
mtlb_diag - Matlab diag emulation function
mtlb_diff - Matlab diff emulation function
mtlb_dir - Matlab dir emulation function
mtlb_double - Matlab double emulation function
mtlb_e - Matlab extraction emulation function
mtlb_echo - Matlab echo emulation function
mtlb_eig - Matlab eig emulation function
mtlb_eval - Matlab eval emulation function
mtlb_exist - Matlab exist emulation function
mtlb_eye - Matlab eye emulation function
mtlb_false - Matlab false emulation function
mtlb_fft - Matlab fft emulation function
mtlb_fftshift - Matlab fftshift emulation function
mtlb_find - Matlab find emulation function
mtlb_findstr - Matlab findstr emulation function
mtlb_fliplr - Matlab fliplr emulation function
mtlb_fopen - Matlab fopen emulation function
mfile2sci
where file is a character string giving the path name of the M-file mfile2sci will generate three files in the same directory
<function-name>.sci: the Scilab equivalent of the M-file
<function-name>.cat: the Scilab help file associated to the function
sci_<function-name>.sci: the Scilab function required to convert the calls to this Matlab M-file in other Matlab M-files. This function may be improved "by hand". This function is only useful for conversion not for use of translated functions.
Some functions like eye, ones, size, sum,... behave differently according to the dimension of their arguments. When mfile2sci cannot infer dimensions it replaces these function call by a call to an emulation function named mtlb_<function_name>. For efficiency these functions may be replaced by the proper scilab equivalent instructions. To get informations about replacement, enter: help mtlb_<function_name> in Scilab command window
Some other functions like plot, has no straightforward Ã©quivalent in scilab. They are also replaced by an emulation function named mtlb_<function_name>.
When translation may be incorrect or may be improved mfile2sci adds a comment which begins by "//!" (according to verbose-mode)
When called without rhs, mfile2sci() launches a GUI to help to select a file/directory and options.
Examples
Done.

Ilustración 79 (Uso SCILAB UMA: Conjunto de funciones previamente emuladas en Scilab)

No todas las instrucciones están soportadas. Por ello, Scilab está provisto un conjunto de herramientas (Ilustración 80) para adaptar estas funciones. Para acceder a estas herramientas tenemos varias vías: o bien utilizando el comando **mfile2sci();** O bien pulsando sobre el icono del *toolbar menú*

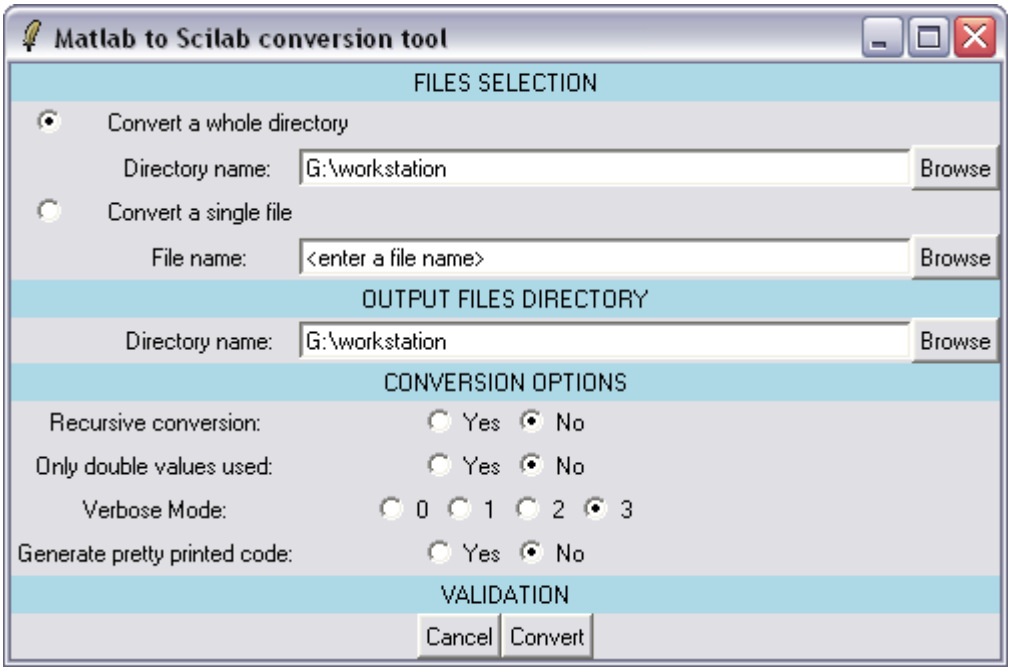


Ilustración 80 (Uso SCILAB UMA: Herramientas para conversión de ficheros de Matlab a Scilab)

Como se muestra en la Ilustración 80 podemos adaptar todos los ficheros de un directorio o un simple fichero, cumplimento el nombre del directorio o el nombre del archivo respectivamente.

Una vez especificado el directorio de salida y las opciones de conversión, pulsamos sobre el botón **Convert.**

Si la traducción de la instrucción es incorrecta o puede mejorarse, mfile2sci añade un comentario que comienza con la cadena de caracteres "//!" (según el nivel de versbose que hayamos especificado).

Nota: Para más ayuda tecleamos **help mfile2sci();**

5.3.7 Modo Consola.

Es posible utilizar el modo no grafico o *modo consola.* Este modo también puede ser llamado modo comandos, ya que al carecer de entorno, solo podemos interactuar introduciendo las sentencias en por teclado (Ilustración 82).

Para activar este modo, tendremos que ir a **Archivo/Edicion/Modo Consola** (Ilustración 81). Tenemos que tener en cuenta que cuando pasamos a *modo consola*, automáticamente se cerrará el entorno gráfico.

Ilustración 81 (Uso SCILAB UMA: Acceso al modo consola)

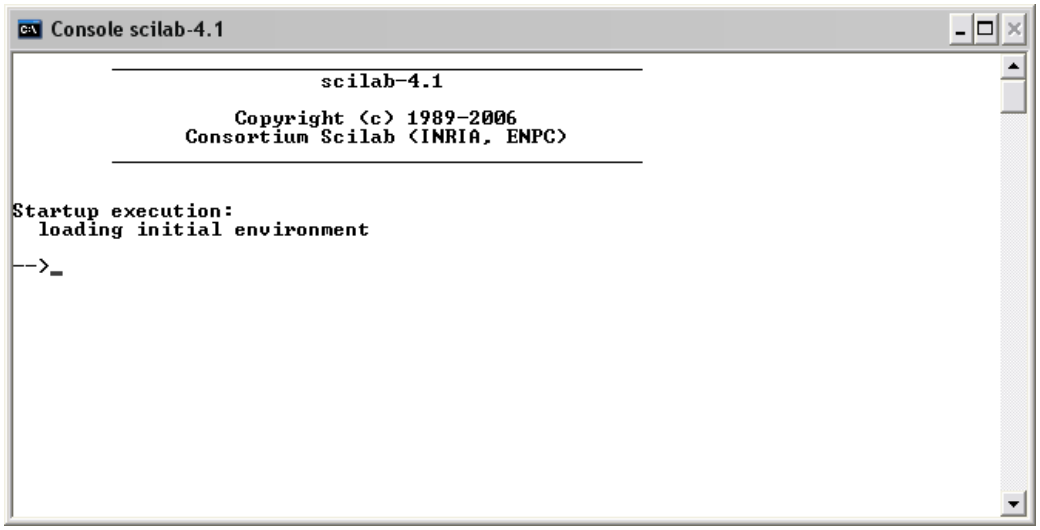


Ilustración 82 (Uso SCILAB UMA: Modo Consola)

Para salir del modo consola, teclearemos la instrucción **exit.**

La consola de comandos nos permite una mayor interactividad y al carecer de entorno grafico los requerimientos se reducen. Como contrapunto podemos decir que este modo es mucho menos intuitivo a la hora de trabajar ya que carecemos entre otros de una ventana con los comandos que vamos tecleando no vemos de forma rápida los comandos, no podemos cambiar tamaño y tipo de letras y sobre todo que no podremos realizar derivación simbólica, ya que esta solo esta implementada en el entorno gráfico.